



JURNAL IPTEK

MEDIA KOMUNIKASI TEKNOLOGI

homepage URL : ejurnal.itats.ac.id/index.php/iptek



Studi Komparatif Model Proses Perangkat Lunak Terhadap Karakteristik Sistem ERP

Rizky Parluka¹, Moch. Fikri Hidayat², Hadiansyah R. Putra³, Vinza Hedi Satria⁴, Hilman F.L.⁵, Faris Hirmar Pralas⁶

Teknik Informatika, Fakultas Ilmu Komputer, UPN "VETERAN" Jawa Timur

INFORMASI ARTIKEL

Jurnal IPTEK – Volume 22
Nomer 2, Desember 2018

Halaman:
1 – 8
Tanggal Terbit :
20 Desember 2018

DOI:
[10.31284/j.iptek.2018.v22i2.252](https://doi.org/10.31284/j.iptek.2018.v22i2.252)

ABSTRACT

A comprehensive and fast response system will be an option for users, but of course requires systematic development unlike simple programs that can be developed independently by individuals. In a sense, this is because the world is facing another information technology revolution which is one of the largest and most complex industries after the industrial revolution in the world. Increased use of the internet, computers and artificial intelligence has changed requirements and practices in business. This paper discusses important issues in the computer world. This is related to a software management process that examines software development through a software process model that will aim to work and affect the quality of software using several development techniques, known as software development cycles. Examples of development models such as waterfall model, iterative, incremental, spiral and extreme programming. All of this model has advantages and disadvantages. Therefore, the main objective of this paper is to examine several software development models and which models can be used to develop ERP systems in accordance with the characteristics of ERP.

Keywords: ERP; Software Engineering Model; Software Engineering; Software Life Cycle.

EMAIL

rizkyparlika.if@upnjatim.ac.id¹
mfhidayat.if@gmail.com²
hadiansyahrp.if@gmail.com³
vinzasatria.if@gmail.com⁴
hilmanfadli.if@gmail.com⁵
farishirmarpralas.if@gmail.com⁶

PENERBIT

LPPM- Institut Teknologi
Adhi Tama Surabaya
Alamat:
Jl. Arief Rachman Hakim
No.100,Surabaya 60117,
Telp/Fax: 031-5997244

*Jurnal IPTEK by LPPM-
ITATS is licensed under a
Creative Commons
Attribution-ShareAlike 4.0
International License.*

ABSTRAK

Sistem yang komprehensif dan memiliki daya respon yang cepat akan menjadi pilihan bagi para pengguna, namun tentu saja memerlukan pengembangan sistematis tidak seperti program sederhana yang dapat dikembangkan secara mandiri oleh perorangan. Sedikit banyak hal itu diakibatkan karena dunia sedang menghadapi revolusi teknologi informasi lain yang merupakan salah satu industri terbesar dan paling kompleks setelah terjadinya revolusi industri di dunia. Meningkatnya penggunaan internet, komputer dan kecerdasan buatan telah mengubah persyaratan dan praktik dalam bisnis. Paper ini membahas isu penting dan penting di dunia komputer. Hal ini terkait dengan proses manajemen perangkat lunak yang meneliti pengembangan perangkat lunak melalui model proses perangkat lunak yang akan memandu kerja dan mempengaruhi kualitas perangkat lunak dengan menggunakan beberapa teknik pengembangan, yang dikenal sebagai siklus pengembangan perangkat lunak. Contoh model pengembangannya seperti pemrograman waterfall, iteratif, incremental, spiral dan extreme. Model ini memiliki kelebihan dan kekurangan. Oleh karena itu, tujuan utama dari disusunnya paper ini adalah untuk menguji beberapa model pengembangan perangkat lunak dan model mana yang dapat digunakan untuk mengembangkan sistem ERP sesuai dengan karakteristik ERP.

Kata kunci: ERP; Model Proses Perangkat Lunak; Rekayasa Perangkat Lunak; Siklus Hidup Perangkat Lunak.

PENDAHULUAN

Setiap aplikasi baik pada komputer dekstop ataupun komputer mobile berjalan melalui perangkat lunak. Teknologi perangkat keras pada komputer banyak mengalami perubahan dan perkembangan akhir-akhir ini, selain perubahan yang signifikan dalam pengembangan perangkat lunak. Biasanya pada tahun-tahun awal, perangkat lunak dikembangkan oleh seorang programmer tunggal atau oleh tim pemrograman kecil sesuai ukuran proyek. Pengembangan ini bergantung pada keahlian programmer dan tidak ada praktik perangkat lunak yang strategis. Fase berikutnya adalah jika ukuran perangkat lunak dan domain aplikasi perangkat lunak meningkat maka akan membutuhkan tim yang lebih besar dan lebih terorganisir. Tujuan dari rekayasa perangkat lunak adalah untuk menciptakan sebuah karya yang sesuai yang membangun program berkualitas tinggi [1]. Untuk itu, ada persyaratan untuk set standar langkah, teknik, dan alat untuk memecahkan masalah ini untuk membangun perangkat lunak, yang disebut model proses perangkat lunak. Hal ini digunakan untuk mengembangkan dan mendukung sistem informasi mereka dan pengembangan sistem ini sering mengikuti siklus pengembangan perangkat lunak [2] [3] [4].

TINJAUAN PUSTAKA

Sistem ERP

Sistem ERP saat ini diminati oleh kedua manufaktur dan organisasi jasa karena mereka menyediakan solusi terintegrasi untuk kebutuhan sistem informasi organisasi [5]. Sistem ERP telah dikenal sebagai sistem yang membawa integrasi ke beberapa aktivitas bisnis di dalam organisasi yang kompleks, sistem ERP menjadi sistem informasi standar, terlepas dari ukuran dan sifat organisasi, sehingga dorongan untuk sistem ERP adalah faktor teknologi dan operasional [6]:

1. Faktor teknologi adalah penggantian sistem yang berbeda (perangkat keras / perangkat lunak) karena sulitnya pemeliharaan, integrasi proses dan sistem, dan perangkat lunak tidak memenuhi kebutuhan bisnis.
2. Faktor operasional adalah kepuasan pelanggan, peningkatan proses, penyederhanaan proses yang kompleks, dan standarisasi proses. Sistem ERP memiliki kemampuan untuk mengotomatisasi dan mengintegrasikan proses bisnis organisasi, untuk berbagi data dan praktik umum di seluruh organisasi, dan untuk memproduksi dan mengakses informasi secara real time [7].

Saat ini, dunia menghadapi revolusi terbesar dan paling kompleks setelah revolusi industri, yang merupakan revolusi teknologi informasi. Meningkatnya popularitas internet, WWW, perkembangan teknologi baru dalam komputasi mobile dan kecerdasan buatan mengubah kebutuhan bisnis dan praktik bisnis. Sistem ERP dapat disesuaikan, perangkat lunak aplikasi standar yang mencakup solusi bisnis terpadu, memberikan peningkatan efisiensi, produktivitas dan kualitas layanan yang signifikan, dan mengurangi biaya layanan serta pengambilan keputusan yang lebih efektif [8].

METODE

Penelitian kami lakukan dengan cara mengumpulkan sebanyak-banyaknya literasi mengenai model-model perangkat lunak yang menjadi objek penelitian pada paper kali ini. Literasi dapat berupa jurnal internasional maupun sebuah artikel. Dari semua literasi ini, kami tarik pengertian, kekurangan dan kelebihan dari beberapa model siklus hidup perangkat lunak menjadi beberapa poin-poin yang akan dijabarkan dibawah. Lalu dari kesimpulan pada literasi itu, kami bandingkan lagi kesimpulan-kesimpulan pada iterasi menghasilkan satu tabel yang mampat.

Model proses perangkat lunak

Model proses perangkat lunak adalah representasi proses yang disederhanakan. Ini menyajikan deskripsi proses dari perspektif tertentu seperti: (1) spesifikasi, (2) desain, (3) validasi, (4) evolusi [1]. Selain itu bentuk-bentuk pembeda dari masing-masing model proses perangkat lunak sebagai faktor pembanding diantaranya adalah: (1) Spesifikasi Kebutuhan, (2) Biaya, (3)

Faktor Resiko, (4) Tingkat Keberhasilan, (5) dan Keterlibatan Pengguna. Pengembangan perangkat lunak akan memandu kerja dan mempengaruhi kualitas perangkat lunak akhir dengan menggunakan beberapa teknik pengembangan. Model ini memerlukan komunikasi yang erat antara klien dan tim pengembangan perangkat lunak untuk meminimalkan kesenjangan pemahaman [9].

Model Waterfall

Waterfall adalah model tradisional dan klasik yang digunakan untuk pengembangan perangkat lunak, Kadang-kadang disebut siklus hidup klasik atau model air terjun, model sekuensial linier, ini adalah model proses pengembangan perangkat lunak pertama yang berasal dari proses rekayasa (Royce, 1970)

Tabel 1. Keuntungan dan Kekurangan Model Waterfall

KEUNTUNGAN	KEKURANGAN
Kemudahan perencanaan dan integrasi, ideal untuk mendukung manajer proyek yang kurang berpengalaman dengan menawarkan cara mudah untuk mengelola dan mengendalikan pekerjaan; kemajuan pengembangan sistem dapat diukur, ketat dalam urutan urutan langkah-langkah pembangunan untuk membantu memastikan kualitas, keandalan, dan pemeliharaan perangkat lunak yang dikembangkan [12].	Tidak mungkin mengumpulkan semua persyaratan saat ini dan masa depan dari pengguna sistem di awal; Diperlukan serangkaian rencana yang ditulis dan disetujui oleh pengguna; Model ini rumit, tidak fleksibel untuk dimodifikasi; sebuah proyek berlangsung memakan waktu, lamban, mahal dan sulit untuk merespons perubahan masa depan dalam siklus hidup; adanya kesenjangan antara spesifikasi dan penyebaran sistem; [11], [15].

Model Iteratif

Kesulitan dengan model air terjun menciptakan tatanan baru untuk memecahkan masalah model air terjun dan yang dapat memberikan hasil yang lebih cepat, memerlukan lebih sedikit informasi terlebih dahulu, dan menawarkan fleksibilitas yang lebih besar. Alamat ini dalam model pengembangan iteratif, ide utama model ini didasarkan pada proyek terbagi menjadi beberapa bagian kecil seperti proses mini waterfall. Hal ini memungkinkan tim pengembang untuk menunjukkan hasil di awal proses dan mendapatkan umpan balik yang berharga dari pengguna sistem [13].

Tabel 2. Keuntungan dan Kekurangan Model Iteratif.

KEUNTUNGAN	KEKURANGAN
Komunitas pengguna terlibat secara aktif sepanjang proyek; proyek ini meruntuhkan bagian-bagian yang lebih kecil karena kurang kompleksnya; dan umpan balik dari pengguna sistem [11].	Komunitas pengguna terlibat secara aktif sepanjang semua fase proyek; Ini memerlukan lebih banyak waktu untuk staf, dan itu berarti menambahkan lebih banyak waktu untuk periode proyek; Model ini membutuhkan lebih banyak keterampilan komunikasi dan koordinasi dalam pengembangan proyek, dan permintaan formal untuk perbaikan proyek setelah setiap fase dapat menyebabkan kebingungan dalam mekanisme yang terkontrol untuk menangani permintaan yang akan dikembangkan; Tidak ada dokumentasi yang tepat untuk proyek saat ini atau masa depan; dan perkembangan cepat sering menyebabkan hasil kualitas buruk [11].

Model Prototipe

Kerangka model ini iteratif, dan mengusulkan untuk berintegrasi dengan model lain seperti air terjun agar lebih efektif. Model prototyping digunakan untuk menampilkan semua kemungkinan sistem aplikasi dan memberikan komunikasi antara pengguna dan pengembang. Fase dari model ini adalah pengembang mendengarkan kebutuhan pelanggan, mengembangkan beberapa demo, membangun kembali dan ini dievaluasi oleh pengguna sampai mereka menerima rancangan dari pengguna. Model ini digunakan bila sulit untuk mengetahui semua persyaratan pada awal proyek [13]

Tabel 3. Keuntungan dan Kekurangan Model Prototipe

KEUNTUNGAN	KEKURANGAN
Lebih efektif daripada model waterfall dalam menghasilkan suatu sistem yang memenuhi kebutuhan pelanggan; memberikan visi awal dari solusi; memungkinkan pelanggan untuk menemukan lebih banyak persyaratan dan berpartisipasi dalam proses; berguna untuk paket kustomisasi; umpan balik yang cepat melintasi aktivitas; klarifikasi pengguna untuk praktik terbaik dari proses desain [16].	Struktur yang buruk; Prosesnya tidak jelas sedangkan sistemnya dikembangkan dengan cepat dan tidak efektif untuk menghasilkan dokumen; membutuhkan alat dan teknik khusus; Sistem yang dirancang dengan buruk dapat menyebabkan sistem kualitas yang cepat dan buruk; bukan integrasi penuh dari sistem.

Model Inkremental

Jenis kerangka adalah kombinasi linear dan iteratif. Ini menggabungkan elemen model air terjun dengan iteratif prototyping. Proses ini mendukung seorang insinyur perangkat lunak untuk mengembangkan perangkat lunak yang lebih komprehensif [13]. Tujuan model ini digunakan untuk mengurangi risiko proyek yang melekat dengan memecah proyek menjadi komponen atau penambahan yang lebih kecil. Fase dalam model ini adalah: komunikasi, perencanaan, analisis, perancangan, implementasi, integrasi, pengujian, dan evaluasi pelanggan. Setiap proses iterasi dan loop umpan balik menghasilkan kenaikan baru sampai produk selesai [9], [16].

Tabel 4. Keuntungan dan Kekurangan Model Inkremental

KEUNTUNGAN	KEKURANGAN
Pelanggan bisa mendapatkan fungsionalitas penting sejak awal, produk awal dikirimkan dengan cepat, lebih fleksibel daripada waterfall dan biaya yang lebih rendah untuk mengubah ruang lingkup dan kebutuhan, membantu mengelola risiko proyek selama pengulangan, penerapan bertahap, produksi lebih cepat, mengurangi pengerjaan ulang dalam proses [11], [13].	Membutuhkan perencanaan dan perancangan yang baik, diperlukan antarmuka yang terdefinisi dengan baik, masalah sulit cenderung didorong sebagai masalah manajemen. Selanjutnya, validasi untuk setiap tahap dan tahap tumpang tindih tidak dipertimbangkan [11],[16].

Model Spiral

Tabel 5. Keuntungan dan Kekurangan Model Spiral

KEUNTUNGAN	KERUGIAN
Analisis yang tinggi untuk mengurangi risiko proyek, memberikan indikasi awal untuk penjaminan kualitas melalui prototipe pada setiap tahap dalam sistem, kombinasi antara model waterfall, prototyping dan incremental sebagai kasus khusus dalam kerangka, kombinasi memberikan umpan balik awal dari pengguna dan menghasilkan perangkat lunak lebih awal. [1], [11].	Biaya tinggi untuk penyesuaian lebih dan cukup kompleks, memerlukan keahlian khusus dalam analisis risiko, tidak sesuai untuk proyek kecil, tidak ada kontrol untuk berpindah dari satu siklus ke siklus lainnya, sehingga setiap fase dapat menghasilkan lebih banyak pekerjaan untuk siklus berikutnya; mereka tidak memiliki tenggat waktu tertentu, juga ada masalah saat persyaratan pengguna tidak jelas [1], [16].

Model Pengembangan Aplikasi Cepat (Rapid Application Development Model - RAD)

Tabel 6. Keuntungan dan Kekurangan Model Rad

KEUNTUNGAN	KEKURANGAN
Untuk melengkapi persyaratan pengguna dengan perkembangan yang cepat, penghematan biaya dan usaha manusia yang kurang, mendorong	Perkembangan cepat dan biaya rendah dapat menyebabkan kualitas buruk; membutuhkan pengembang dan desainer

KEUNTUNGAN	KEKURANGAN
umpan balik pelanggan, pelanggan berpartisipasi sepanjang siklus yang lengkap, memungkinkan penyesuaian paket, menggunakan konsep pemodelan untuk mencapai tingkat fleksibilitas dan kemampuan perawatan yang tinggi, memberikan integrasi pada awalnya, mengelola perubahan dalam desain sistem seperti yang diminta oleh pengguna [11], [16].	yang sangat terampil; sulit digunakan dengan sistem warisan; risiko untuk tidak pernah menutup masalah proyek; banyak kebutuhan untuk partisipasi pelanggan; sulit dengan penggunaan ulang modul untuk sistem masa depan; tidak cocok untuk proyek yang lebih murah dan untuk proyek keselamatan dan kritis. [11], [16].

Model Pengembangan Aplikasi Cepat (Rapid Application Development Model - RAD)

Reuse Oriented Model adalah sistem yang merakit dari komponen yang ada. Hal ini juga telah dipandang sebagai konsep untuk pengembangan sistem yang cepat. Model seperti ini disebut Components Based System (CBS) yang membangun atau mengembangkan aplikasi perangkat lunak dari komponen perangkat lunak atau komponen yang ada. Komponen ini direkatkan untuk menyusun aplikasi. Ide dasar di balik model reuse adalah perangkat lunak yang dikembangkan di beberapa bagian, komponen perangkat lunak ini disebut komponen. Selain itu, setiap komponen merupakan bagian dari sistem untuk melengkapi fungsi dan dirancang untuk melayani tujuan tertentu [13]. Fase dari model ini adalah: persyaratan, analisis dan perancangan, pengembangan dan pengujian, dan fase pelepasan dan pengelaran [17].

Tabel 7. Keuntungan dan Kekurangan Model ROM

KEUNTUNGAN	KERUGIAN
Pengiriman perangkat lunak yang lebih cepat ke organisasi: mengurangi biaya dan risiko: mengurangi jumlah perangkat lunak yang akan dikembangkan: Ini memberikan efisiensi dan kehandalan.	Tidak memenuhi kebutuhan nyata pengguna karena mengurangi persyaratan adalah suatu keharusan: kehilangan beberapa kontrol atas evolusi sistem: terbatas untuk digunakan dalam lingkungan pengembangan berorientasi objek dan kinerja rendah merupakan masalah utama

Model Pengembangan Aplikasi Cepat (Rapid Application Development Model - RAD)

Extreme Programming (XP) adalah metodologi pengembangan perangkat lunak tangkas yang menangani perubahan yang cepat dalam persyaratan dan lingkungan secara bertahap dan siklus rilis. XP juga dapat digambarkan sebagai metode pengembangan untuk proyek yang tidak semua persyaratan dan fitur sistem dapat didefinisikan dan direncanakan sebelum pengembangan dimulai. Tahapan model XP tercantum di bawah sebagai berikut:

Tabel 8. Keuntungan dan Kekurangan Model XP

KEUNTUNGAN	KERUGIAN
Pengembangan cepat dan penghematan biaya, menghasilkan kohesi tim dan kooperator tim yang lebih baik, penekanan pada produk akhir, terintegrasi dengan metode pengembangan tradisional, proses iteratif, berarti jarak pendek antara rilis sistem, kesederhanaan dan umpan balik yang cepat, pengujian berdasarkan persyaratan dan kualitas. Jaminan.	Perkembangan yang cepat menyebabkan kualitas perangkat lunak yang buruk, membutuhkan pengalaman dan keterampilan, mahal dengan menggunakan beberapa alat sebagai pasangan pemrograman (dua pemrogram), sulit memprioritaskan perubahan, menghabiskan waktu antara pelanggan dan tim pengembang akan menghasilkan kesuksesan, dan kebutuhan konstruksi kasus uji mengkhhususkan keterampilan.

HASIL DAN PEMBAHASAN

Melalui literasi yang telah di teliti oleh kami, kami bandingkan bermacam-macam model proses perangkat lunak ke dalam 7 poin yang menjadi karakter ERP sebagai berikut :

Tabel 9. Perbandingan Model Proses Perangkat Lunak

Karakter ERP	MODEL PROSES PERANGKAT LUNAK							
	Waterfall	Iteratif	Prototipe	Spiral	Inkremen	RAD	ROM	XP
Integritas	Tinggi	Sedang	Rendah	Rendah	Sedang	Rendah	Sedang	Sedang
Kekompleksan	Tinggi	Sedang	Sedang	Sedang	Sedang	Sedang	Sedang	Rendah
Kelengkapan	Tinggi	Rendah	Rendah	Rendah	Rendah	Rendah	Sedang	Rendah
Fleksibilitas	Rendah	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Rendah	Tinggi
Penerapan Terbaik	Sedang	Tinggi	Tinggi	Tinggi	Tinggi	Sedang	Sedang	Tinggi
Paket Perangkat Lunak	Rendah	Sedang	Rendah	Rendah	Sedang	Rendah	Rendah	Tinggi
Keterbukaan	Sedang	Sedang	Rendah	Rendah	Rendah	Rendah	Sedang	Rendah

KESIMPULAN

Kesimpulan berikut dapat ditarik dari pembahasan paper ini yaitu bahwa dari delapan model proses perangkat lunak yang dibahas diatas, masing masing memiliki keunggulan dan kekurangan sendiri dalam penerapannya. Namun dari kedelapan model diatas, model yang tidak terlalu mengedepankan fleksibilitas hanyalah model waterfall dan ROM. Dan model yang dapat diterapkan lebih baik dalam pengaplikasiannya adalah model iteratif, model prototipe, model spiral, model inkremen, model XP (Extreme Programming). Setiap model memiliki kelebihan dan kekurangan tersendiri. Dari hasil analisis seperti terlihat pada tabel evaluasi, pengembang dapat memilih pengembangan perangkat lunak model siklus hidup sesuai dengan kebutuhan.

UCAPAN TERIMA KASIH

Kami ucapkan terima kasih kepada seluruh pihak yang telah membantu kami dalam pembuatan Jurnal ini yaitu pihak UPNV JATIM dan juga sanak keluarga yang turut memberi support kepada kami selama pembuatan jurnal ini dan juga pihak ITATS yang telah memberikan kami kesempatan untuk melakukan publish jurnal. Terima kasih juga kepada Bapak Nanang Fakhrrur Rozi selaku editor dari pihak ITATS yang telah memberikan panduan terhadap format jurnal Itats

DAFTAR PUSTAKA

- [1] Akhilesh. "Assessment of Software Process Models." International Journal of Engineering and managment Sciences, vol. 3, no. 4, 2012: 500-506.
- [2] Hoffer, A. Jeffery. Modern System Analysis and Design (Sixth Edition ed). Upper Saddle River, New Jersey 07458: Pearson Education, Inc, 2011.
- [3] Pressman, R." Software Engineering A Practitioner's Aproach, New York: McGraw-Hill" 2010.

-
- [4] Dahalin, Z., Abd Razak, R., Ibrahim, H., Yusop, N. and Kasiran, M.K. "An Enterprise Architecture Methodology for Business-IT Alignment: Adopter and Developer Perspectives." *Communications of the IBIMA*, vol.2010, (2011): 1-14.
 - [5] Shehab, E. M., M. W. Sharp, and T. A. Spedding. "Enterprise resource planning An integrative review." *Business Process Management Journal* 10, no. 4 (2004): pp. 359-386.
 - [6] Okunoye, Adekunle, Frolick Mark, and Crable Elaine. "ERP Implementation in Higher Education: An Account of PreImplementation and Implementation Phases." *Cases on information technology* 8, no. 3 (2006): 110-133.
 - [7] Nah, Fiona Fui-hoon, Janet Lee-shang Lau, and Jinghua Kuang. "Critical factors for successful implementation of enterprise systems." *Business Process Management Journal* 7, no. 3 (2001): 285-296.
 - [8] Ioannoua, G., and C. Papadoyiannisab. "Theory of ConstraintBased Methodology for Effective ERP Implementation." *International Journal of Production Research* 42, no. 23 (2004): 4927-4954.
 - [9] Khan, U.A. "Improved Iterative Software Development Method for Game Design. University of Tampere, Department of Computer Sciences." 2008.
 - [10] Souza, C. and Zwicker, R. "ERP System' Life Cycle: An Extended Version." *Technology-Idea group Inc.*, 2009: 1426-1431.
 - [11] Sasankar, A. and Chavan, V. "SWOT Analysis of Software Development Process Models." *International Journal of Computer Science Issues*, vol. 8, no. 5, (2011): 390-399.
 - [12] Ratbe, Dina, William R King, and Young-gul Kim. "the fit between project characteristics and application development methodologies." *The Journal of Computer Information Systems*, 2000.
 - [13] Qureshi, M, and S Hussain. "A reusable software component-based development process model." *Advances in Engineering Software* 39, no. 2 (2008): 88-94.
 - [14] Sommerville, I. *Software Engineering* (8th Edition ed.). Addison Wesley , 2006.
 - [15] Brehm, L. "Tailoring ERP Systems : A Spectrum of Choices and their Implications." *Proceedings of the 34th Hawai'i International Conference on System Sciences*. 2001.
 - [16] Services, O.I. "Selecting a development approach." *Office of Information Services, centers for medicare & medicaid*, (2008): 110.
 - [17] He, Y. "A Comparative Study of Critical Success Factors for ERP System Implementation in China and Finland", (2007).
 - [18] Roger Pressman, titled "Software Engineering - a practitioner's approach".
 - [19] David Alex Lamb. *Software Engineering: Planning for Change*. Prentice-Hall, 1988.
 - [20] Alan M. Davis. *Software Requirements: Objects, Functions, and States*. Prentice Hall PTR; 2nd Revised edition (March 1993).
 - [21] A comparative study of different life cycle models in different scenarios," *international journal of advanced research journal in computer science and management studies*", vol.1, issue5, Oct 2013.
 - [22] Javanovich ,D.Dogsa.T, *Proceedings of 7th international conference on 11-13, June 2003*, pp.582-592.
 - [23] Laura C. Rodriguez Martinez, Manuel Mora ,Francisco,J.Alvarez, "A Descriptive/Comparative Study of the Evolution of Process Models of Software Development Life Cycles", *Proceedings of the 2009 Mexican International Conference on Computer Science IEEE Computer Society Washington,DC, USA*, 2009.
 - [24] Jovanovich, D., Dogsa, T., "Comparison of software development models," *Proceedings of the 7 th International Conference on, 1113 June 2003, ConTEL 2003*, pp. 587-592.
 - [25] Seema , SonaMalhotra , "comparative analysis of popular SDLC models ", *International Journal of advances in computing and information technology*, July 2012, pp.277-286.

Halaman ini sengaja dikosongkan